

FIG. 1

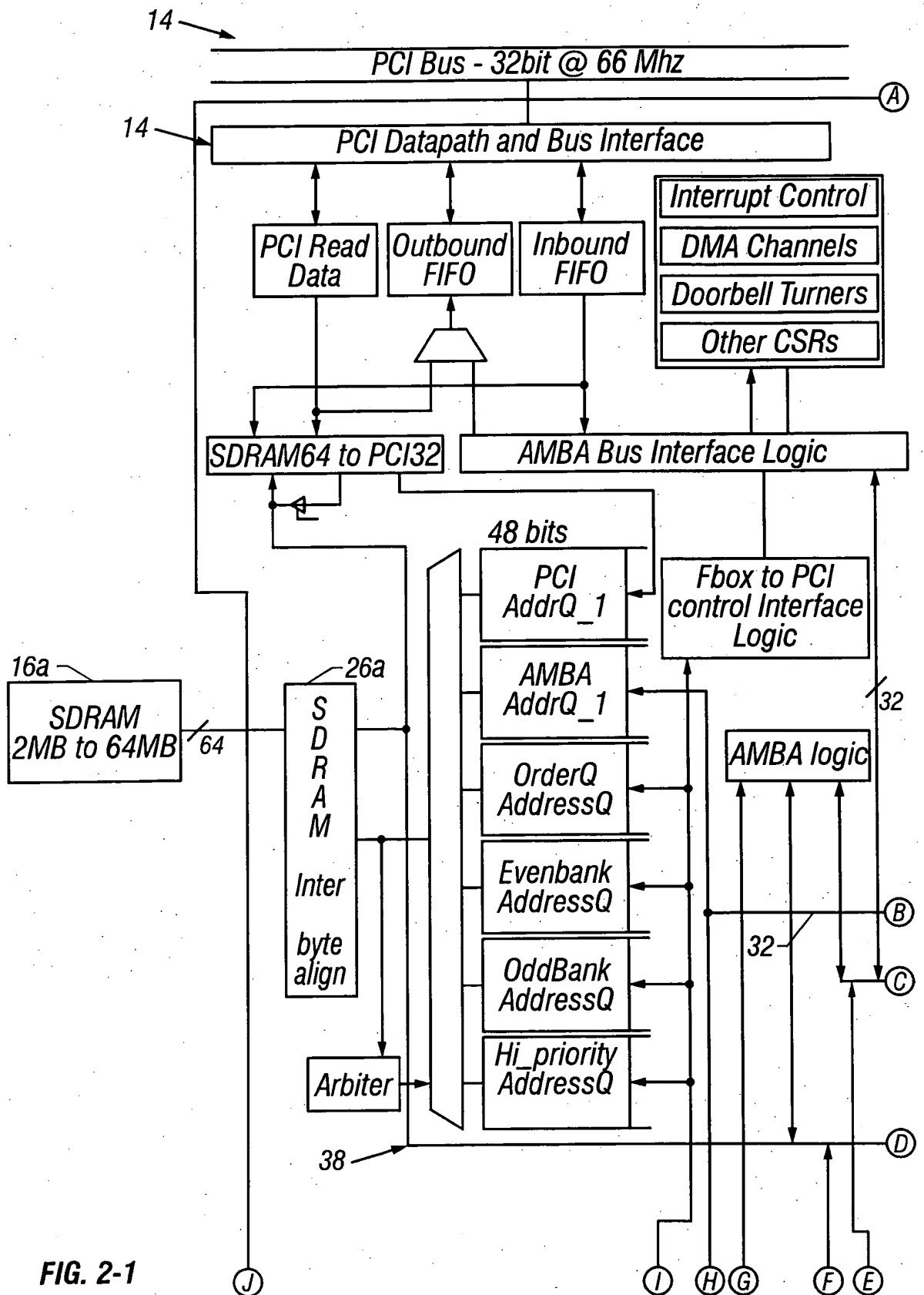
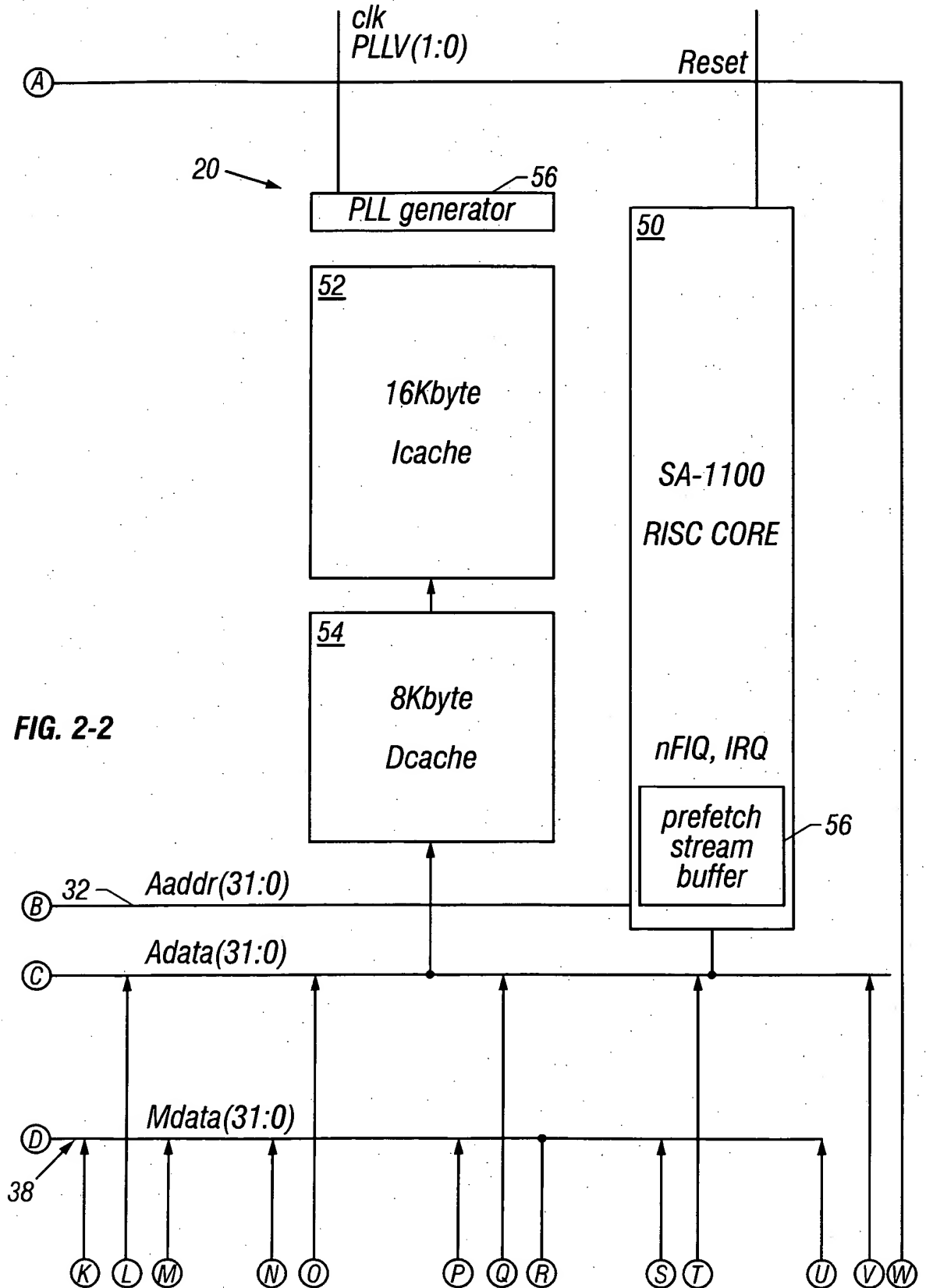


FIG. 2-1



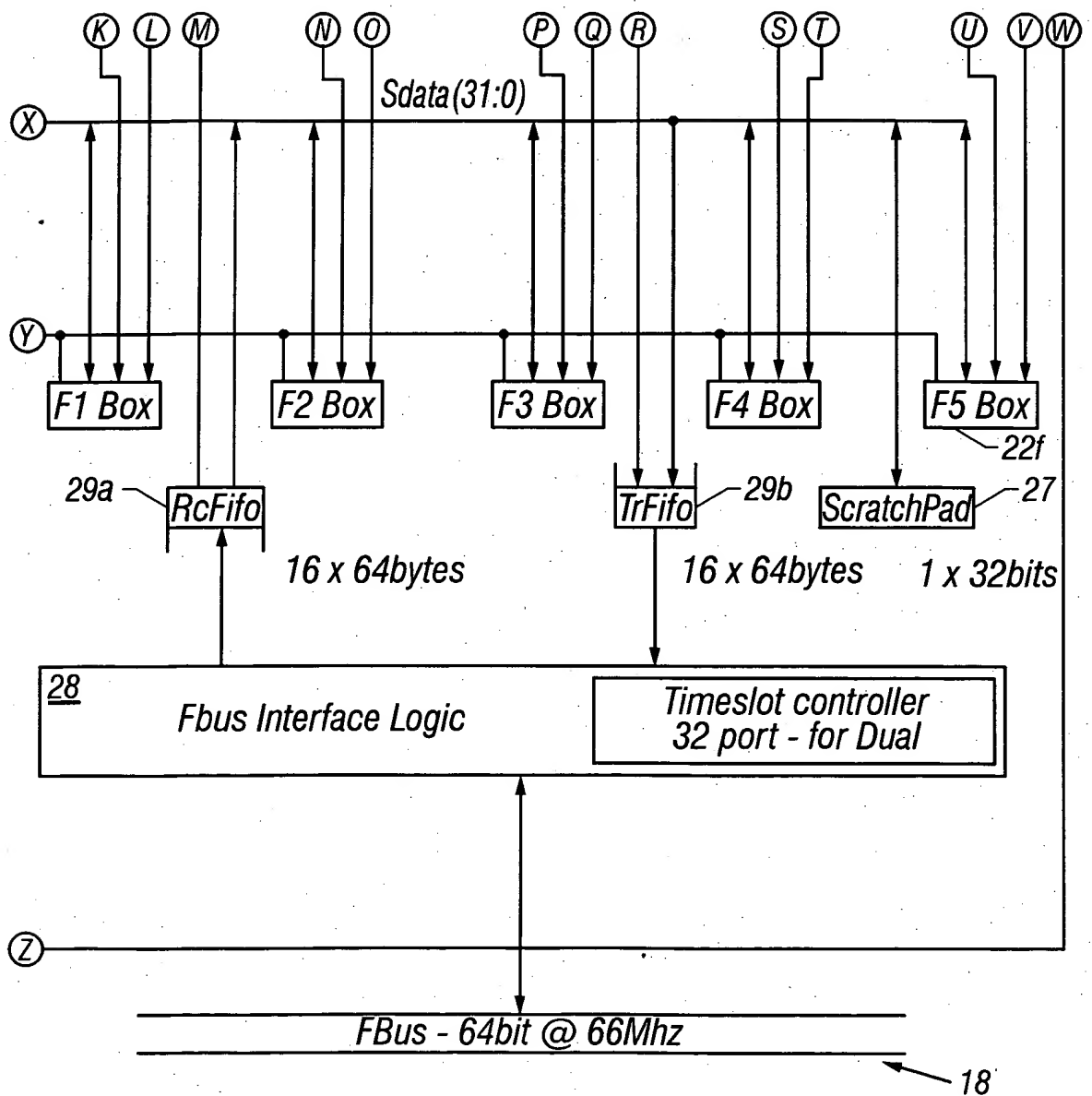


FIG. 2-4

AMBAI31:01

MBUSI31:01

SBUSI31:01

SEQ#_event_response
 FBI_event_response
 sram_event_response
 sdram_event_response
 amba_event_response

Context Event Arbiter

uengine controller

uPC_1

uPC_2

uPC_3

uPC_4

decode

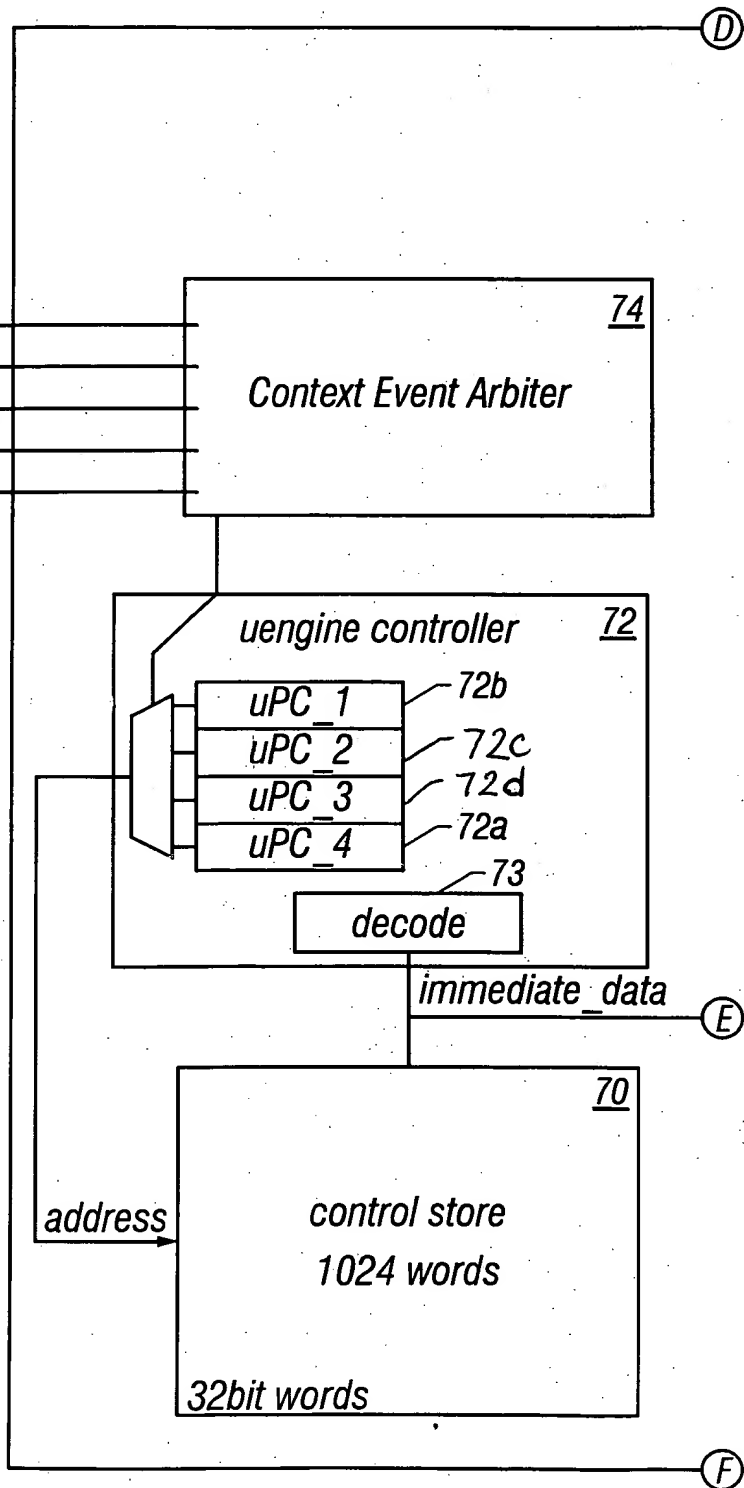
immediate_data

address

control store
 1024 words

32bit words

FIG. 3-1



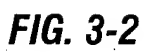


FIG. 3-2

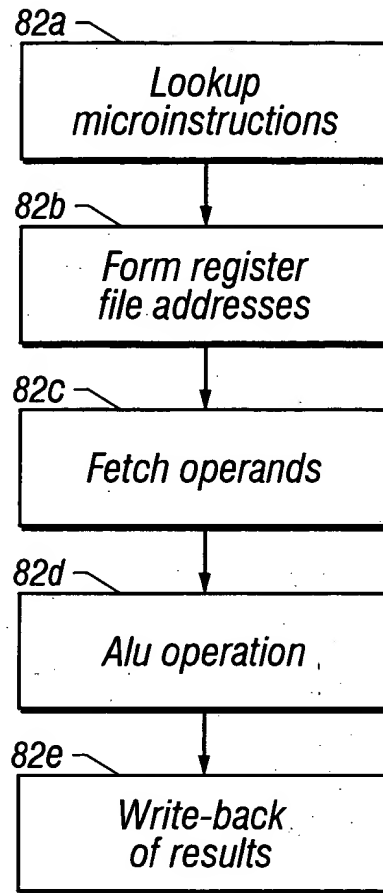


FIG. 3A


```
iva1XXXXXXXXX ctx cmd
```

5) *cxt_cmd*

FIG. 3B

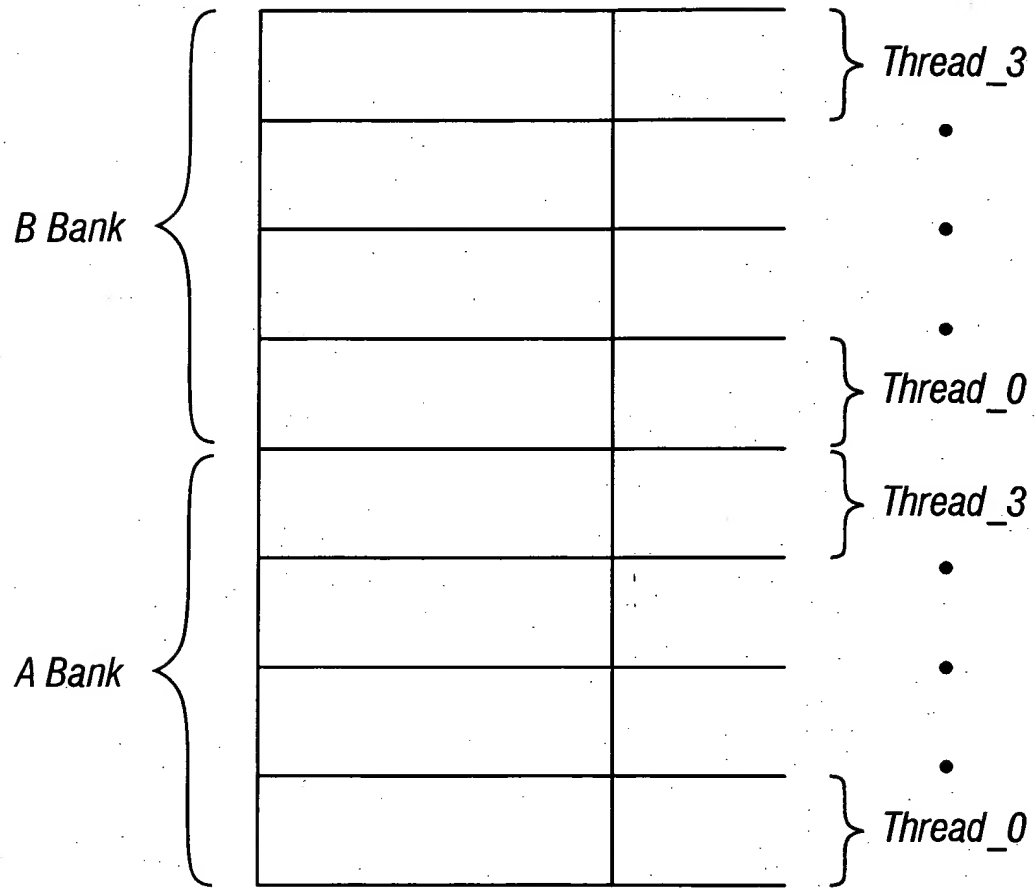


FIG. 3C

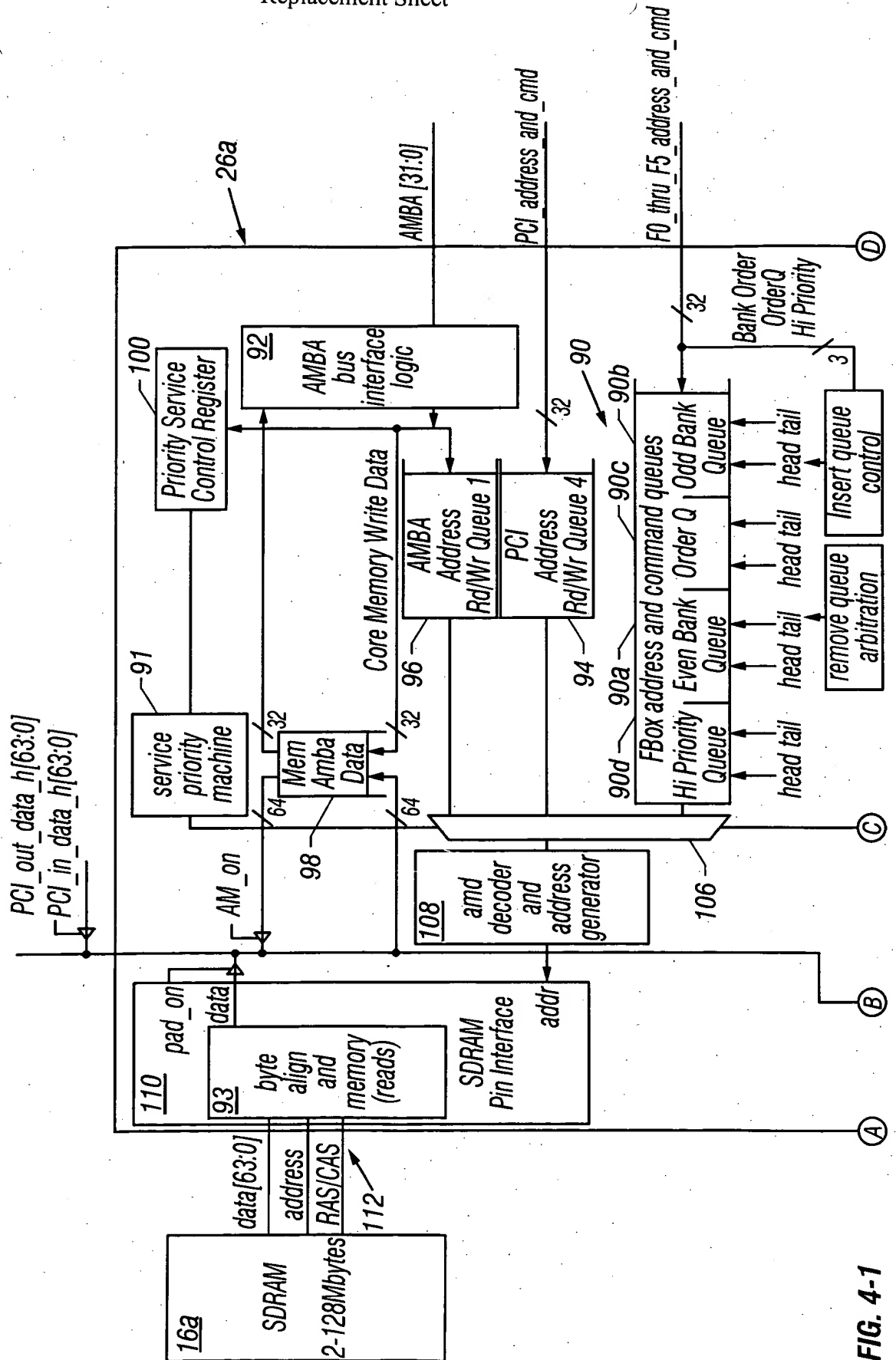


FIG. 4-1

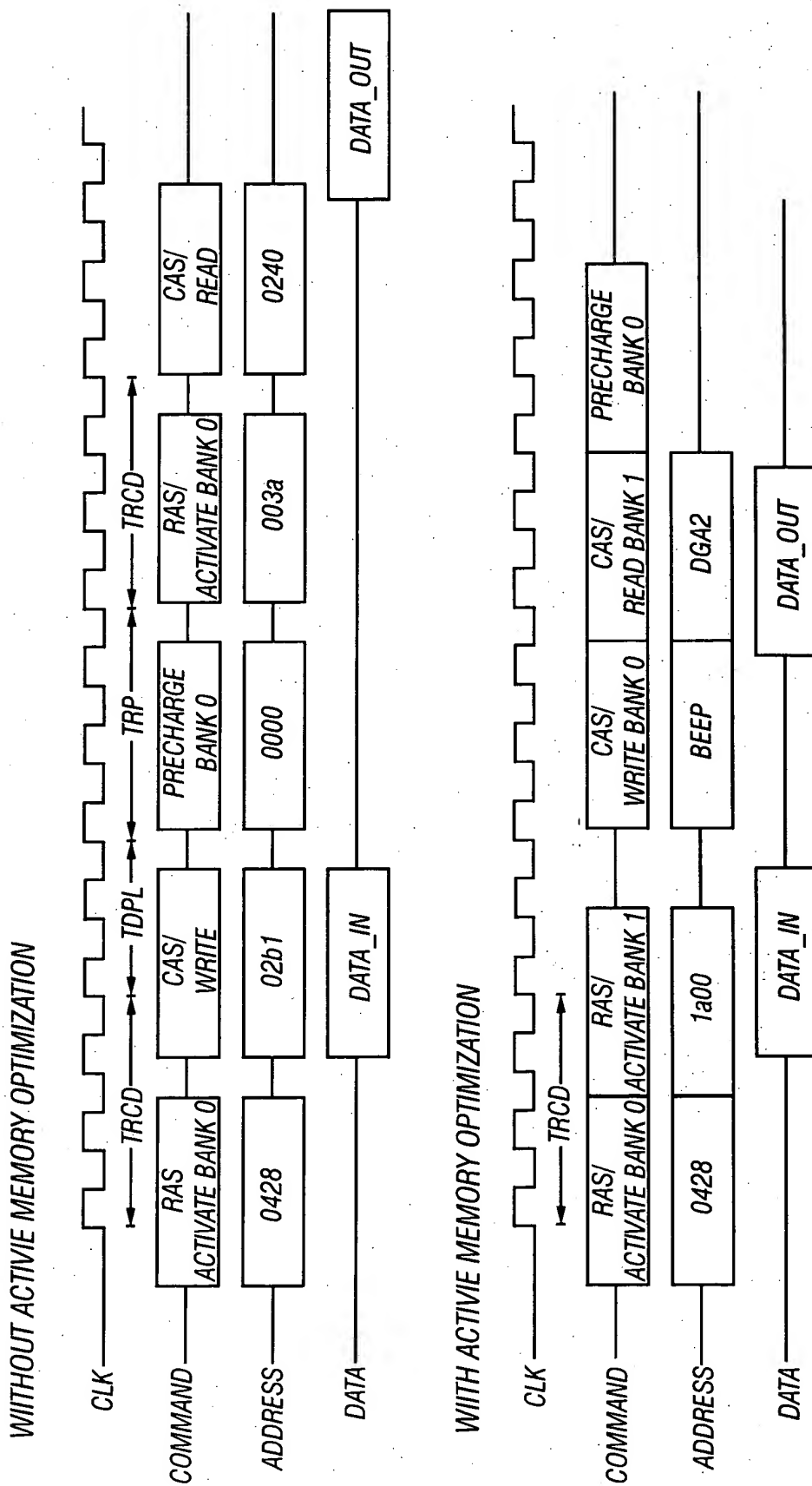
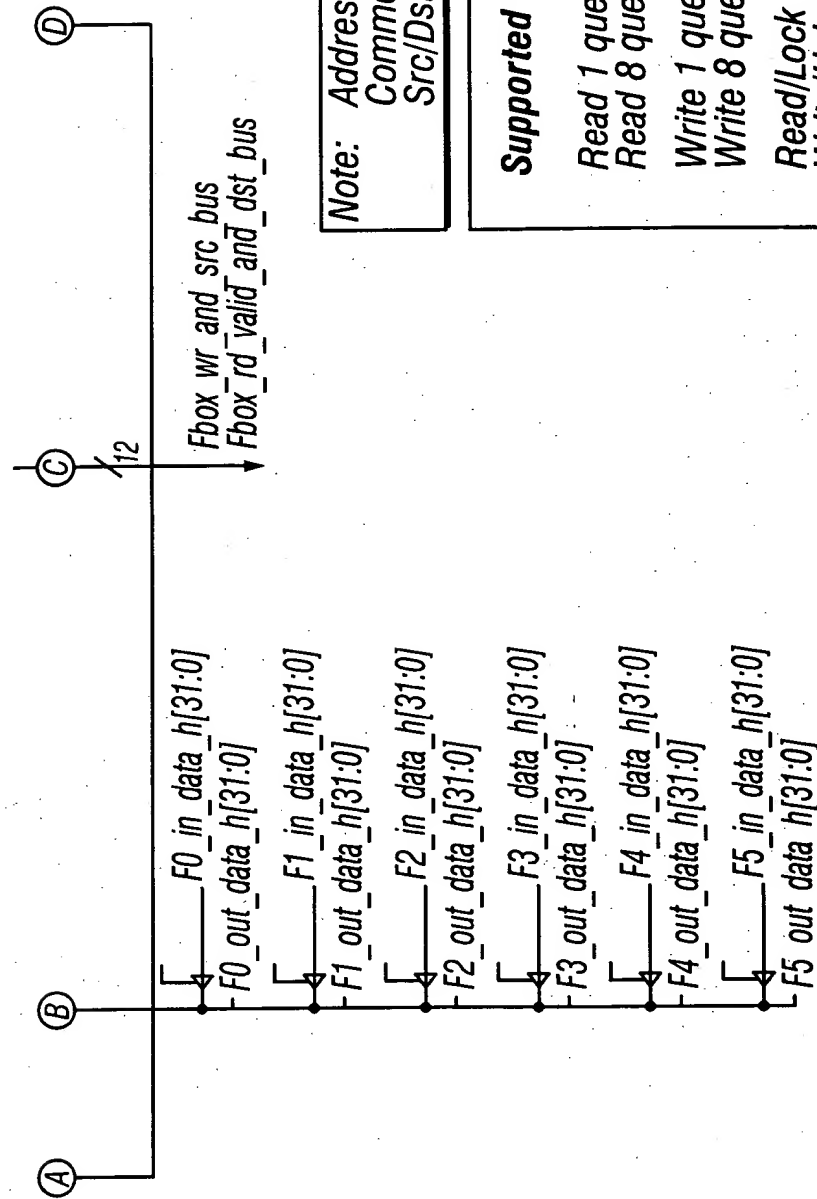


FIG. 4B



Note: Address: 22 bits
 Comment: 4 bits
 Src/Dst: 6 bits

Supported Comments:

- Read 1 questword
- Read 8 questwords
- Write 1 questword
- Write 8 questwords
- Read/Lock
- Write/Unlock
- Unlock
- Reserve n locks
- Insert Queue Element
- Remove Queue Element

FIG. 5-2

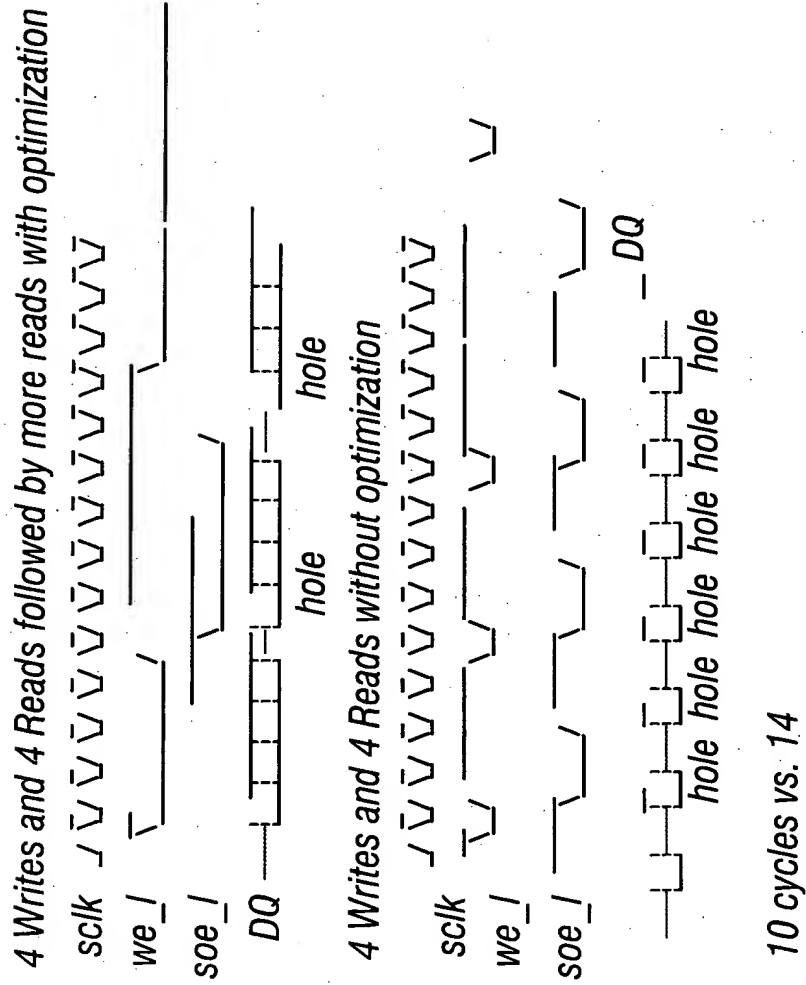


FIG. 5A

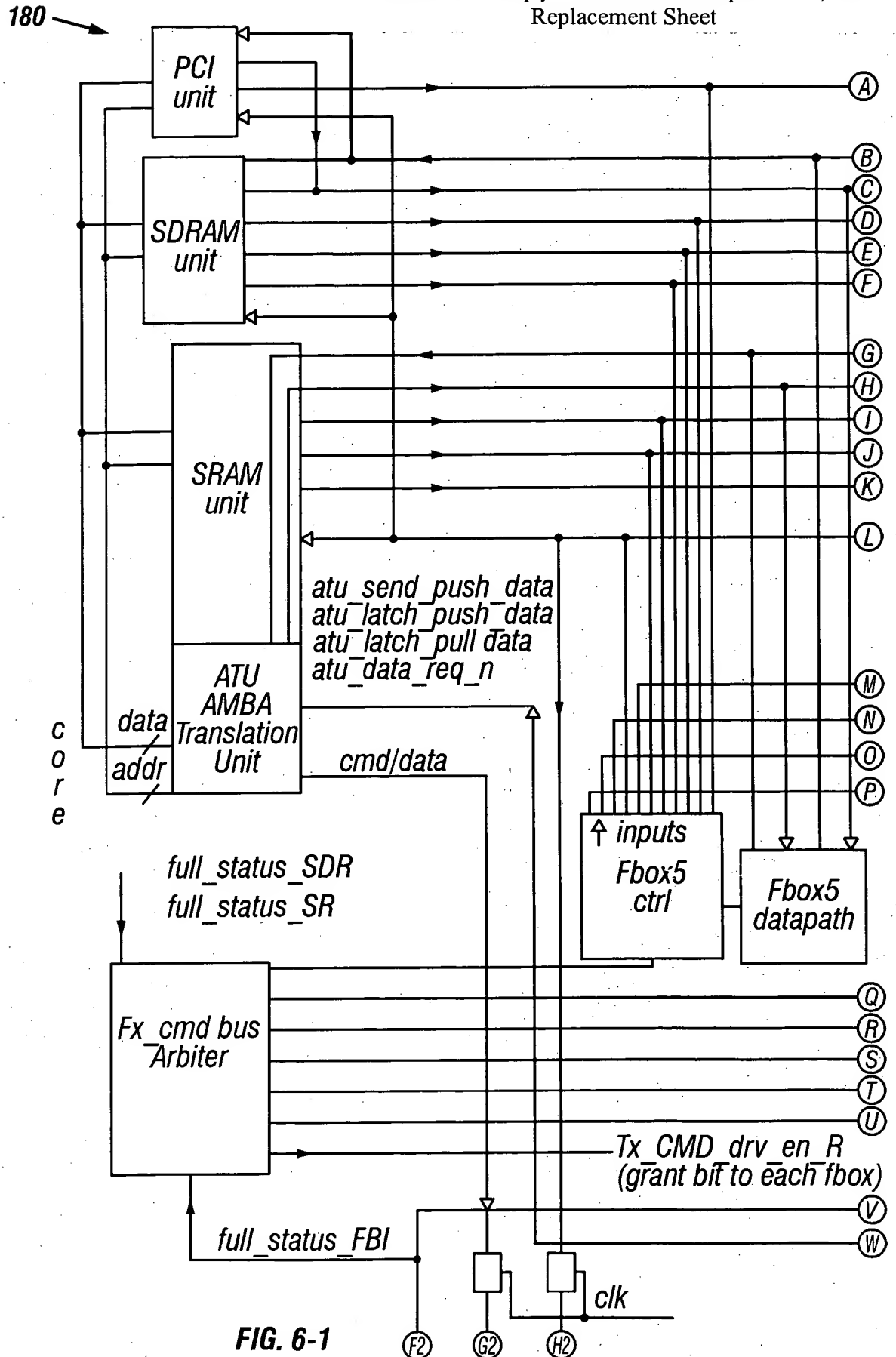


FIG. 6-1

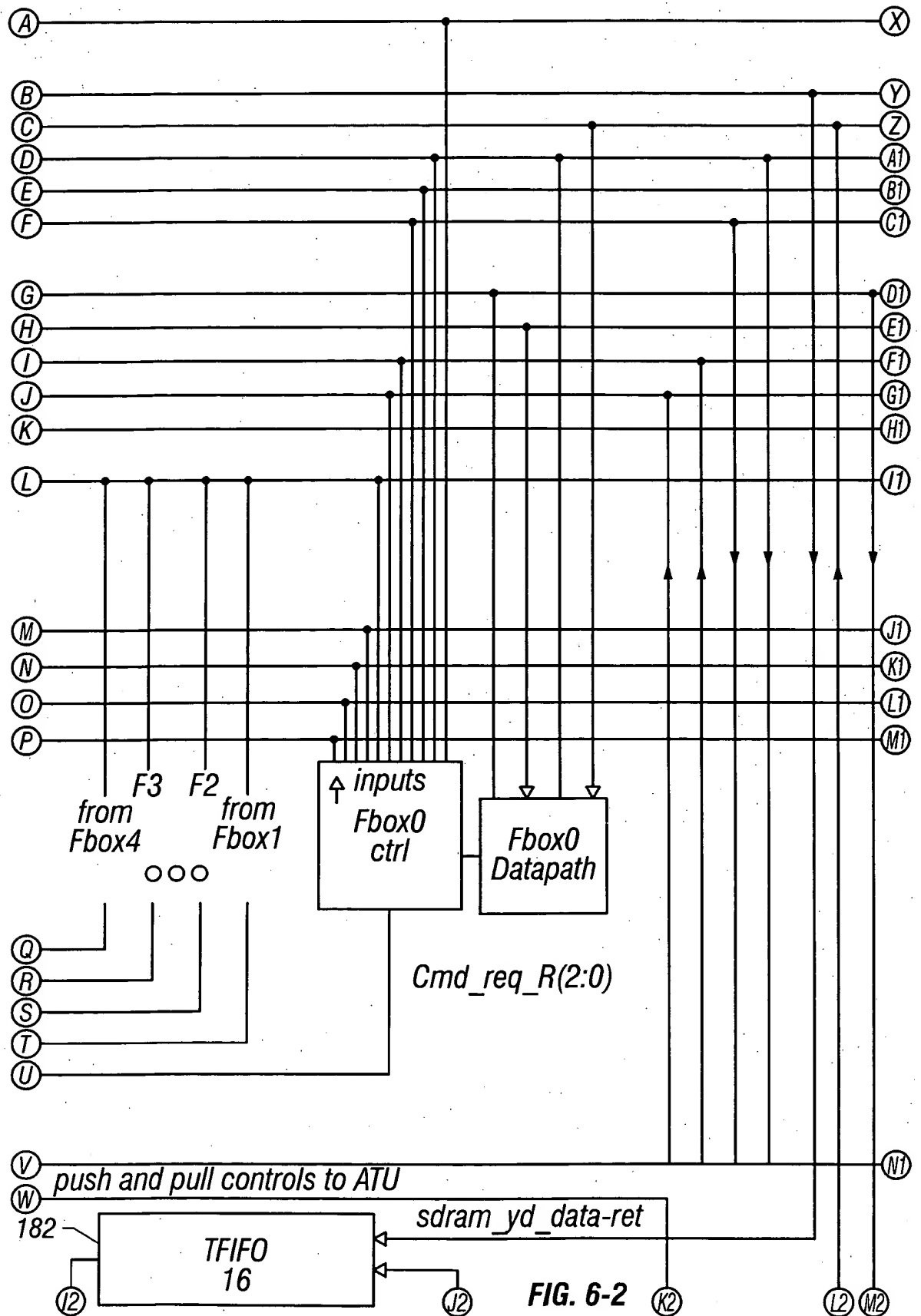


FIG. 6-2

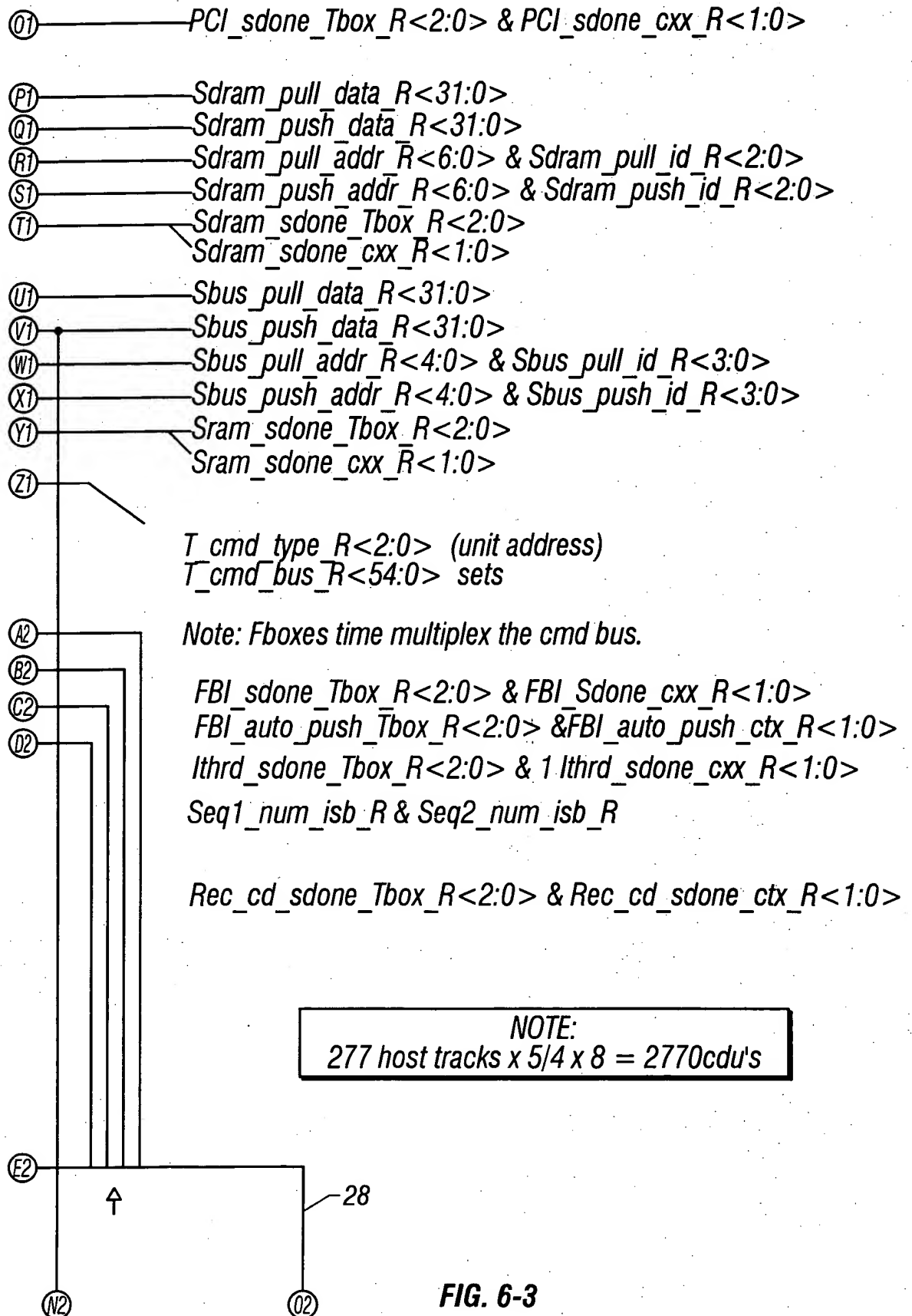
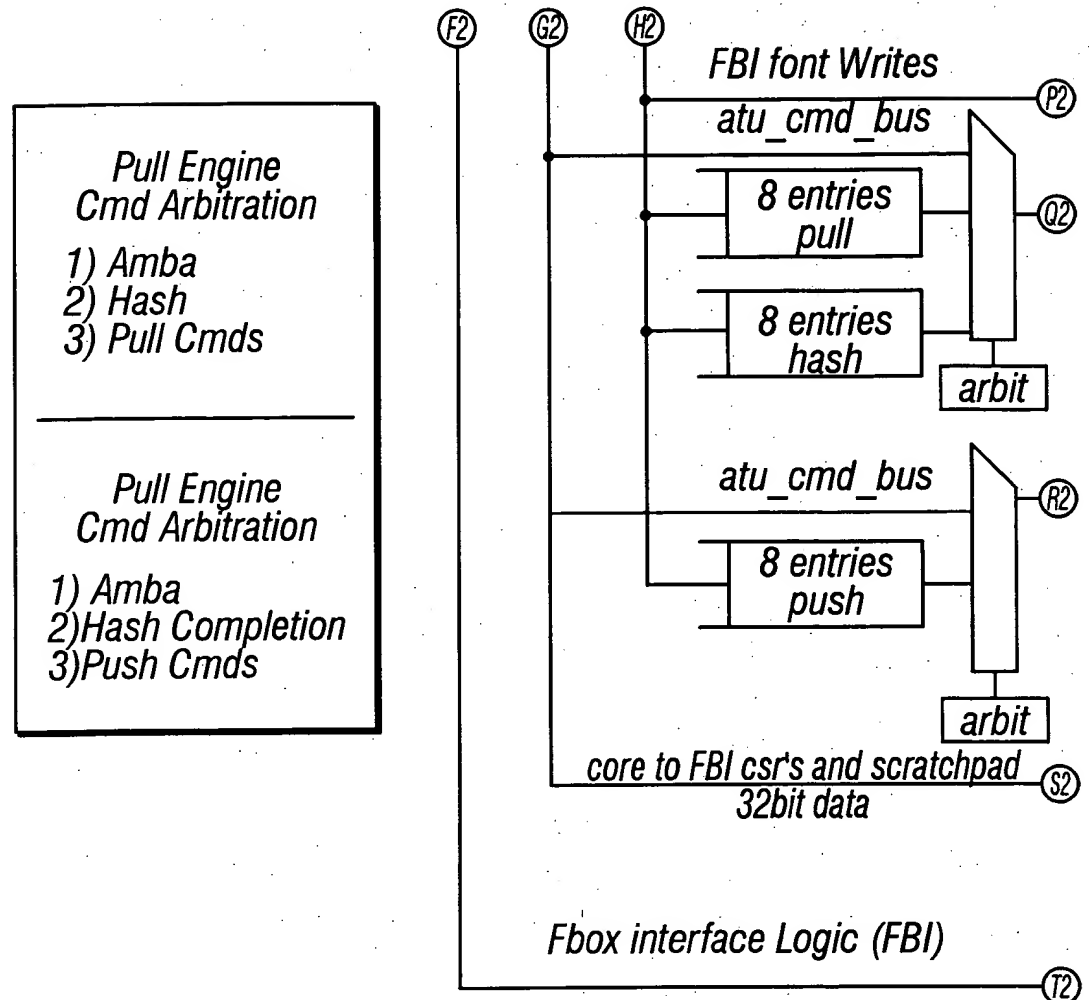


FIG. 6-3



ATU Notes:

- a) Core to FboxRegs:
use sram_push_data_bus
- b) Core to FBI Regs:
use private ATU/FBI
cmd/data bus
- c) Core reads FboxRegs:
use SRAM_pull_data_bus
- d) Core reads FBIRegs:
use sram_push_data_bus
(makes sram appear like
another Fbox to FBI on
sram_push_bus)

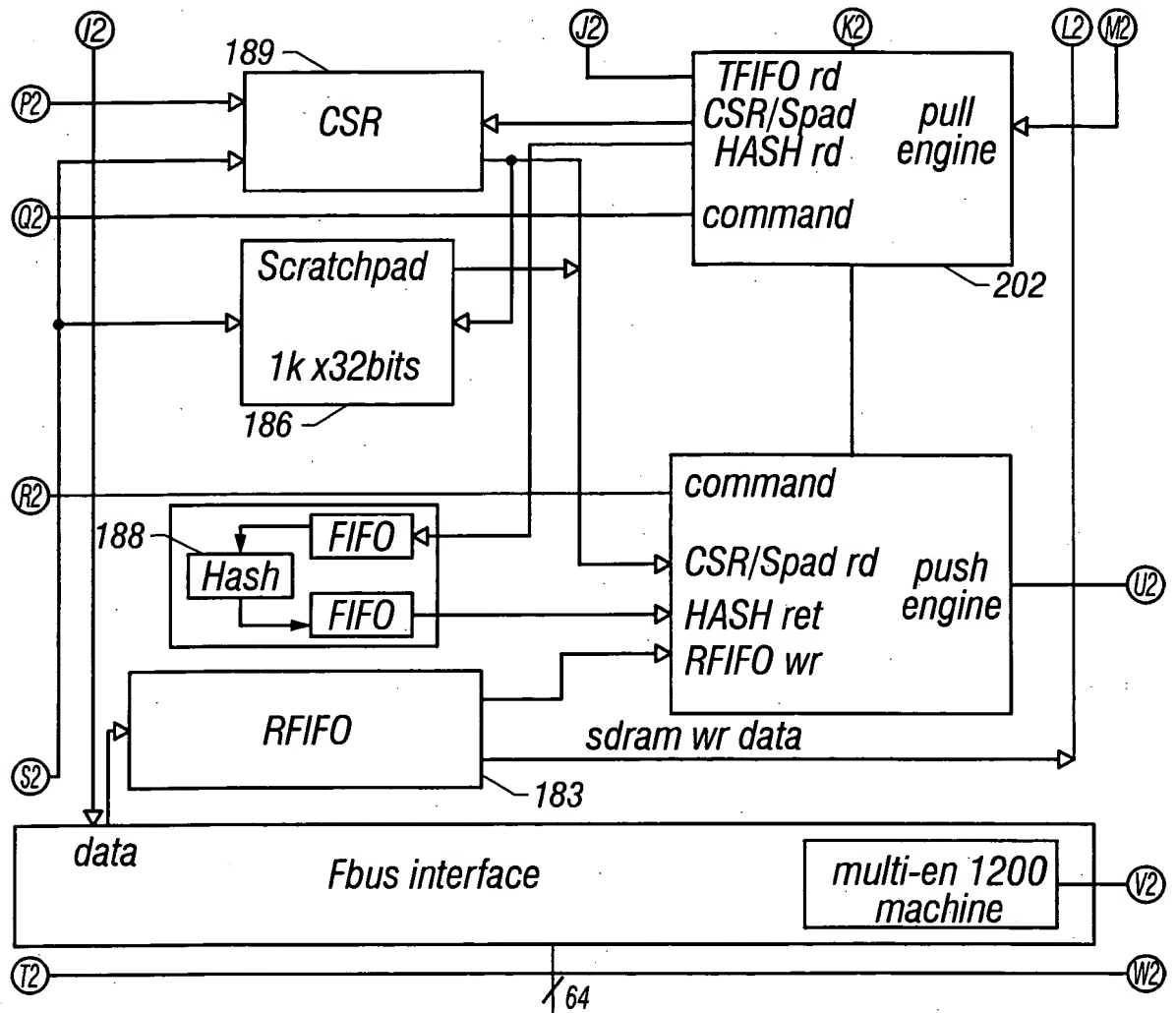
Cmd_Req_R<2:0>

- 000 none
- 001 Sram Chain
- 010 SDR chain
- 011 Sram
- 100 SDR
- 101 FBI
- 110 PCI
- 111

Tx_CMD_drv_en_R<1:0>

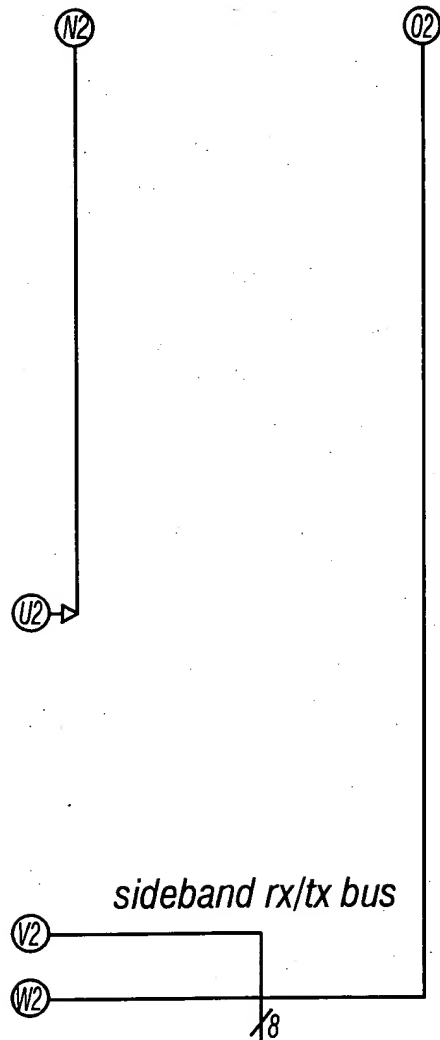
- 0 none
- 1 grant

FIG. 6-4



Sdram_puXX_addr_R<6:0> [4:0] xfer_reg_addr if not TFIFO [6:0] TFIFO_addr	Sram_puXX_addr_R<4:0> [4:0] xfer_reg_addr
Sdram_puXX_ID_R<3:0> 0-5 Fboxes 8-13 Fboxes-csr 6 fbi 15 nop	Sram_puXX_ID_R<3:0> 0-5 Fboxes 8-13 Fboxes-csr 6 fbi 15 nop

FIG. 6-5



T_Cmd_type_R<2:0>

000: bus idle
 001: SDRAM
 010: SRAM
 011: SRAM-csr
 100: PCI
 101: reserved
 110: FBI
 111: Scratch

Fbox Branch/Ctx Choices

1) FBI_sdone	<i>br / ctx</i>
2) FBI_auto_push	<i>br / ctx</i>
3) lthread_sdone	<i>br / ctx</i>
4) signal_rec_cxt	<i>br / ctx</i>
5) Seq#1_change (flag)	<i>br / ctx</i>
6) Seq#2_change (flag)	<i>br / ctx</i>
7) SRAM_sdone	<i>br / ctx</i>
8) SDRAM_sdone	<i>br / ctx</i>
9) volunteer_cxx_swap	<i>ctx</i>
10) Rec_req_available (flag)	<i>br</i>
11) SDRAM_rd_parity_en (flag)	<i>br</i>
12) Fbox_push_protect	<i>br</i>
13) ccodes, contexts and kill	

FIG. 6-6